

# BlueField w/ DOCA Flow

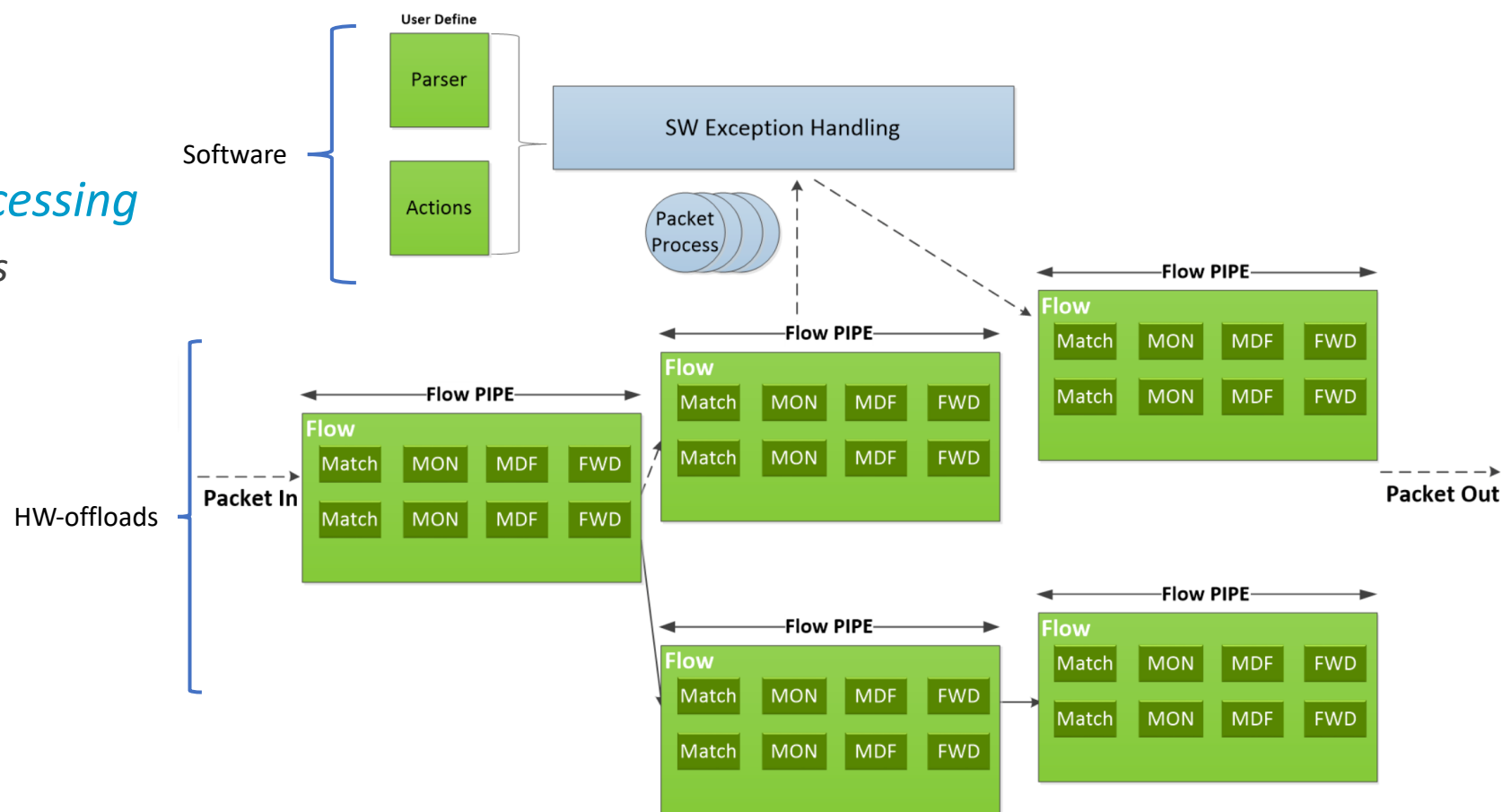
*Guanshujie Fu & Prof. Jialin Li*



# DOCA Flow<sup>1</sup>

DOCA HW offload flow library

- *Pre-req*
  - **DPDK<sup>2</sup>**
- *Offer HW-acc for packet processing*
  - Forward/Drop/Modify packets
  - Match/Monitor
  - Strip/Add tunnel (en/decap)
- *Architecture*
  - Flow -> Pipe(s) -> Entry(s)

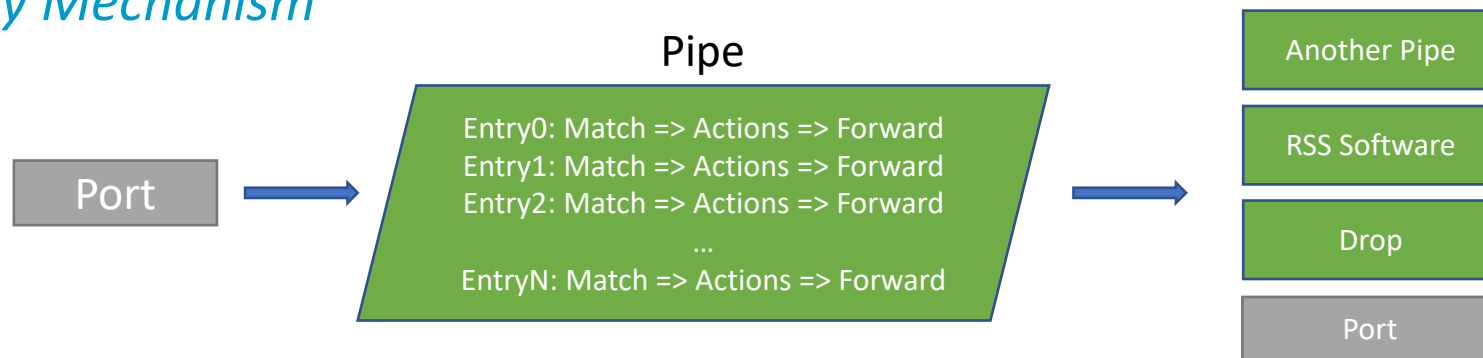




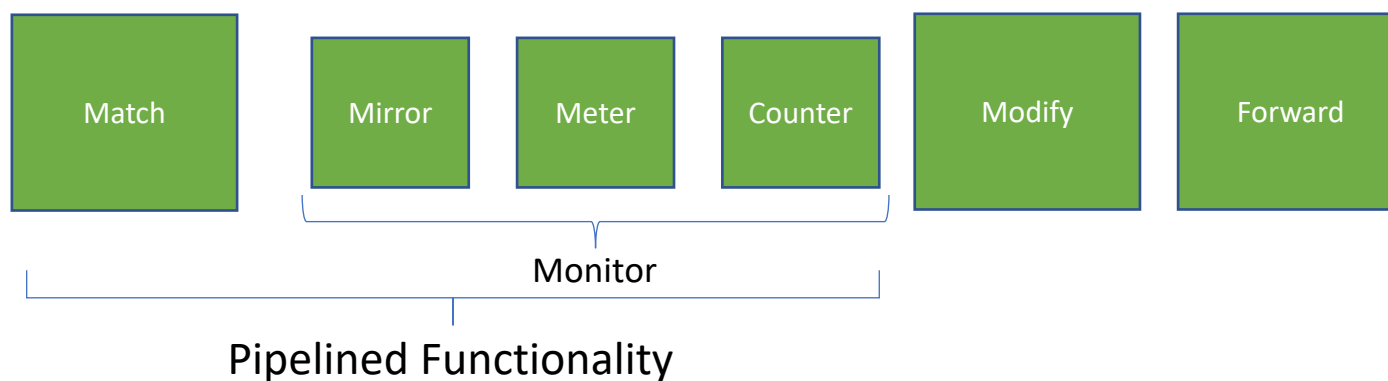
# DOCA Flow Cont'd

DOCA HW offload flow library

## ■ Key Mechanism



## ■ Pipe entry Structure<sup>1</sup>



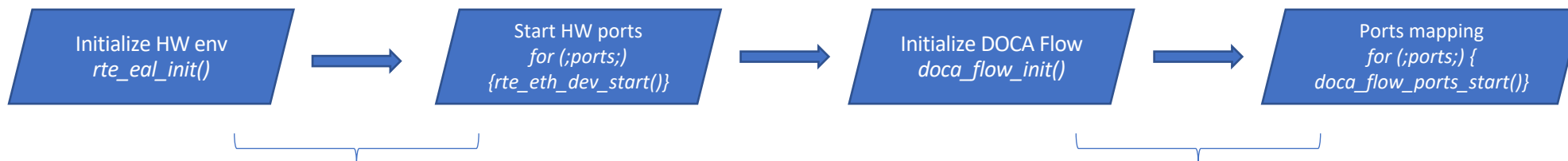


# DOCA Flow Cont'd

DOCA HW offload flow library

## ■ *Programming Philosophy<sup>1</sup>*

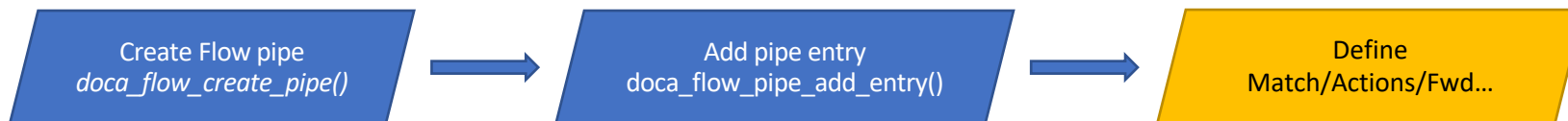
### ■ *HW Initialization*



**DPDK PMD HW Initialization**

**DOCA Flow Initialization**

### ■ *Flow Initialization*



### ■ *Tearing Down (after application terminates)*





# DOCA Flow Cont'd

## DOCA HW offload flow library

### ■ Match

- struct `doca_flow_match`<sup>1</sup>
  - **Match on** MAC/IP/L4/Metadata/...
  - Both in/out packets
- the user-defined fields that should be matched on the pipe

### ■ Actions

- struct `doca_flow_actions`<sup>2</sup>
  - **Modify** MAC/IP/L4
  - Encap/decap data
- struct `doca_flow_action_desc`<sup>3</sup>
  - Define the action type

### ■ Forward

- Next Pipe/Port/RSS Software/Drop

#### Public Variables

```
uint32_t flags
struct doca_flow_ip_addr in_dst_ip
    uint8_t in_dst_mac[DOCA_ETHER_ADDR_LEN]
    doca_be16_t in_dst_port
    doca_be16_t in_eth_type
    uint8_t in_l4_type
struct doca_flow_ip_addr in_src_ip
    uint8_t in_src_mac[DOCA_ETHER_ADDR_LEN]
    doca_be16_t in_src_port
    uint8_t in_tcp_flags
    doca_be16_t in_vlan_tci
struct doca_flow_meta meta
struct doca_flow_ip_addr out_dst_ip
    uint8_t out_dst_mac[DOCA_ETHER_ADDR_LEN]
    doca_be16_t out_dst_port
    doca_be16_t out_eth_type
    uint8_t out_l4_type
struct doca_flow_ip_addr out_src_ip
    uint8_t out_src_mac[DOCA_ETHER_ADDR_LEN]
    doca_be16_t out_src_port
    uint8_t out_tcp_flags
    doca_be16_t out_vlan_tci
struct doca_flow_tun tun
```

```
DOCA_FLOW_ACTION_AUTO - modification type derived from pipe action
DOCA_FLOW_ACTION_CONSTANT - modify action field with the constant value from pipe
DOCA_FLOW_ACTION_SET - modify action field with the value of pipe entry
DOCA_FLOW_ACTION_ADD - add field value. Supports meta scratch, ipv4_ttl, ipv6_hop, tcp_seq, and tcp_ack.
DOCA_FLOW_ACTION_COPY - copy field
```

1. [https://docs.nvidia.com/doca/sdk/doca-libraries-api/annotated.html#structdoca\\_\\_flow\\_\\_match](https://docs.nvidia.com/doca/sdk/doca-libraries-api/annotated.html#structdoca__flow__match)

2. <https://docs.nvidia.com/doca/sdk/flow-programming-guide/index.html#doca-flow-actions>

3. <https://docs.nvidia.com/doca/sdk/flow-programming-guide/index.html#doca-flow-action-desc>



# Applications w/ DOCA Flow

Samples Provided by *NVIDIA*

## ■ *Simple Forward*<sup>1</sup>

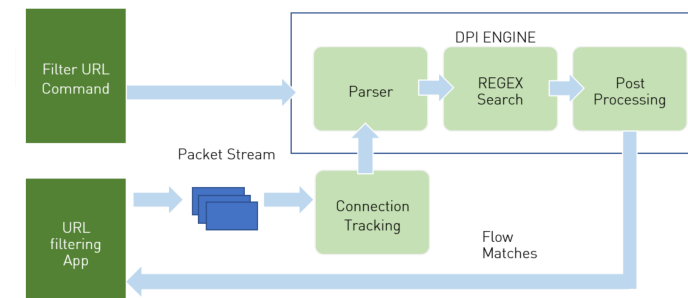
- a forwarding application that takes either VXLAN, GRE, or GTP traffic from a single RX port and transmits it on a single TX port

## ■ *Switch*<sup>2</sup>

- a network application that leverages the DPU's hardware capability for ***internal switching between representor ports*** on the DPU

## ■ *URL Filter*<sup>3</sup>

- limits access by comparing web traffic against a database to prevent users from different threats, malware and accessing harmful sites such as phishing pages



1. <https://docs.nvidia.com/doca/sdk/simple-forward-vnf/index.html>
2. <https://docs.nvidia.com/doca/sdk/switch/index.html>
3. <https://docs.nvidia.com/doca/sdk/url-filter/index.html>



# Sketchy

## TCP w/ DOCA Flow

### ■ Integrate **DOCA Flow** into the TCP stack

- offload partial functionality of TCP network stack, like congestion control, into Bluefield
- utilize Arm core and HW-acceleration on DPU

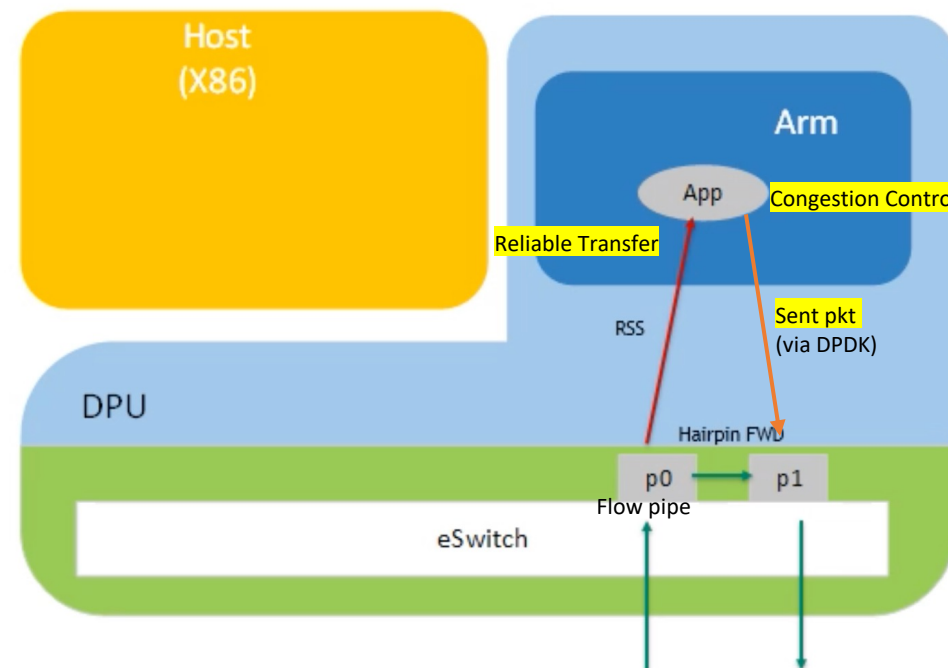
### ■ Motivation

- potential supports for packet process application
- like network stack
- Tonic system<sup>1</sup>

### ■ High-level Idea

in *vnf* mode, packets can be sent to Software on Arm via RSS

- use DOCA Flow to provide hw-acc for packet process
  - **DOCA\_FLOW\_ACTION\_ADD**
  - Support tcp\_seq/tcp\_ack modification
- use Arm core software for reliable transfer/congetst ctrl....
  - Packets will be stored in DPDK queue
  - Software read packets from the queue, and proceed





# Problems

Issues encountered or worth considering

- ***DOCA Flow is highly dependent on DPDK***
  - *need solid knowledge of DPDK*
  - *maybe directly using DPDK to develop the network stack will be a better approach*
- ***Limited supports provided by DOCA Flow***
  - *DOCA Flow only provides limited supports on packet process*
- ***Current scheme is not so compatible with the use of DOCA Flow***
  - *do not fully utilize the hw-acc features*
  - *develop can be tough: no previous experience can be referenced to*
- ***Any further questions?***
  - *I will try my best to answer*



# Thank you

*Guanshujie Fu & Prof. Jialin Li*